A Case for Open Mission Systems in DOD Aircraft Avionics

Capt Michael J. Brown, USAF
Robert D. Fass, PhD
Jonathan Ritschel, PhD

Disclaimer: The views and opinions expressed or implied in the Journal are those of the authors and should not be construed as carrying the official sanction of the Department of Defense, Air Force, Air Education and Training Command, Air University, or other agencies or departments of the US government. This article may be reproduced in whole or in part without permission. If it is reproduced, the Air and Space Power Journal requests a courtesy line.

Background

Leaders within the DOD are challenged to maintain global military supremacy while they simultaneously confront the competing goals of cost efficiency and technological superiority. Without a deliberate strategy designed to maximize efficiency within the Defense Acquisition System, neither goal can be met without compromising the other. If a policy could be implemented that simultaneously reduced costs and encouraged the development of cutting-edge technology, leaders within the DOD could ensure that global superiority is met at an acceptable price to the American taxpayer. The open systems architecture (OSA) concept is intended to meet this challenge.

Given the rapidly changing nature of technology, an acquisition approach that incorporates the ability to upgrade a component without upgrading the entire system is crucial. In essence, this approach encapsulates the intent of OSA—[1] a systems engineering approach focused on employing modular designs, hardware interface standards, and common software and software reuse to ensure openness and component interoperability.[2]

Open mission systems (OMS) is a subset of the OSA approach specifically designed for the military aviation environment. Adopted after two years of collaboration between the DOD and the industry, OMS seek to leverage competition by requiring prime contractors to use DOD-owned and controlled open standards for their avionics suite. The suite enables other companies to produce components that will work on the prime contractor's platform.[3] This requirement is similar to what Apple has done to allow programmers to develop applications that work on the OS operating system. This approach aims to avoid dependency on sole-source providers and reduce monopoly power by allowing avionics systems to be upgraded modularly by several contractors. In theory, the approach effectively leverages competition to reduce the schedule and cost of an avionics upgrade. Since the DOD and industry have agreed on the OMS standards, the costly redesign of the entire avionics suite due to the former proprietary nature of the integration of components is theoretically avoided.

While the OSA is commonplace in commercial industry as demonstrated by Apple, Android, Microsoft, and others, incorporating the principles into DOD aircraft avionics systems is inherently more complicated due to the nature of the product—a fighter aircraft is fundamentally more complex than a cell phone. Nevertheless, the benefits of an open approach have not been overlooked within the DOD. Although the complexity of avionics systems has made the successful adoption of an OMS, or an OMS-like, avionics suite extremely difficult, the DOD has been working toward this goal since the PAVE PILLAR program—a third-generation avionics suite designed for advanced tactical fighter aircraft like the F-22 in the late 1980s. Third-generation avionics architectures were the first to provide a standardized architecture. Similar to OSA in conceptual form, third-generation avionics architectures incorporated a modular approach that allows the replacement of individual modules without replacing the entire avionics system. The third-generation avionics architectures used common hardware and software with a goal of decreasing life-cycle costs for upgrades and repair. They used common data processors that integrated sensor information providing enhanced capabilities to the user. While these practices were successful and a step in the right direction, third-generation architectures were still proprietary and closed, forcing reliance on an obsolete processor that limited upgrade potential.[4] Although third-generation avionics incorporate common standards and a modular approach, the design prevents rapid evolution and competition within the industry due to the closed nature of the architecture. Commercially based hardware and software could not be leveraged, forcing reliance on proprietary hardware and software for any maintenance or upgrade.[5]

The recent collaboration between the DOD and industry to establish OMS is a result of the failed attempts in the past to produce a truly open and modular design. While it took the better part of two decades to come to fruition, OMS was formally adopted on 30 April 2014 and has been successfully implemented and demonstrated on a small scale much to the pleasure of key leaders within the DOD.[6] Successful implementation, however, does not address the other theoretical OMS benefit of reduced costs and schedule. Thus, the purpose of this analysis is to examine data from historical non-OMS aircraft avionics that are analogous to the OMS-enabled demonstration platform to determine whether software development and test costs and the associated schedule decrease with the OMS approach.

Data

A limited number of tests and demonstrations of the OMS concept have been conducted. As of July 2015, Lockheed Martin successfully integrated and flight-tested seven payloads adhering to OMS standards on the U-2 Dragon Lady.[7] Northrop Grumman has also conducted OMS-related tests on the RQ-4 Global Hawk.[8] While the reception from the various tests has been positive, access to data to examine the claims that OMS will reduce the costs and schedule of upgrades in DOD avionics systems is extremely limited. Currently, the full data set required to conduct the analysis (integration cost and schedule data, as well as associated source lines of code) is limited to only one OMS-enabled aircraft platform. As a result, this analysis is exploratory in nature. Nevertheless, available data are enough to garner preliminary insight as to whether or not the OMS approach is a promising method for avionics upgrades in the future.

To conduct the comparative analysis, data are needed on both OMS and the analogous non-OMS avionics upgrade programs.[9] Source documents for the non-OMS analogous avionics upgrade programs include Software Resources Data Reports (SRDR) and Cost Analysis Requirements Documents (CARD) to match the costs of selected historical avionics modernization programs with specific software coding effort measured in source lines of code (SLOC). Data pertaining to the sole OMS-enabled demonstration platform was provided by a Secretary of the Air Force-level cost agency.

The comparison data set consists of 16 historical avionics upgrade programs that were deemed analogous by Secretary of the Air Force subject matter experts to the avionics components replaced in the OMS demonstration platform. Of the 16 historical programs, several were achieving an upgrade for the same platform and were therefore combined. After the appropriate programs were combined, 13 programs remain to compare against the OMS demonstration platform. The data set also contains cost data for the initialization of the OMS demonstration platform and a subsequent OMS upgrade for the platform.

Between the data set—SRDRs, and CARDs—the costs and software fall into two general categories: (1) mission processor software (including Operational Flight Program [OFP] and associated costs; and (2) platform integration software (including application interfaces [API] to avionics applications) and associated costs.

Of the 13 historical programs, eight contain both cost and SLOC data for Category 1, 10 contain both cost and SLOC data for Category 2, and six contain complete data for both categories. One program in the data set contains data for SLOC and research and development (R&D) months but no cost data for either category. Table 1 summarizes the availability of the OMS demonstration platform data and the historical programs' data.

Table 1. Data on analogous programs and OMS demonstration platform

Asset	*Cat 1: Mission Processor/OFP*	*Cat II: Platform Integration/API*	*Cat I: Mission Processor/OFP*	*Cat II: Platform Integration/API*	*Research and Development*
	Costs	*Costs*	*SLOC*	*SLOC*	*Months*
Historical Program 1	X	X	X	X	X
Historical Program 2		X	X	X	X
Historical Program 3	X	X	X	X	X
Historical Program 4	X	X	X	X	
Historical Program 5		X	X	X	X
Historical Program 6	X	X	X	X	X
Historical Program 7	X	X	X	X	X
Historical Program 8	X	X	X	X	
Historical Program 9		X		X	X
Historical Program 10				X	X
Historical Program 11		X	X	X	X
Historical Program 12	X		X		X
Historical Program 13	X		X		X
OMS Demo	X	X	X	X	X
OMS Demo Upgrade	X	X	X	X	X

Data Analysis Results

To compare the OMS-enabled asset to the historical programs, the data are normalized to mitigate the differences in scope and functional complexity between the programs. All SLOC counts are normalized to equivalent SLOC (ESLOC) using a formula provided by a Secretary of the Air Force cost agency through their previous analyses of SRDRs and historical data.[10]

Estimators use ESLOC to determine the effort needed to complete a program. The implicit assumption is that costs increase as ESLOC increases because greater ESLOC should be indicative of greater effort. This relationship is tested in our data set through regression analysis. Specifically, we examine the relationship between the cost data of the historical programs and the corresponding ESLOC of the programs. The results indicate a moderate positive relationship.[11] This lends statistical backing to the belief that as software coding efforts increase (as measured by ESLOC), costs also increase.

With the relationship between cost and ESLOC demonstrated, the next step is comparative analysis. The basic question to answer is: Which approach is more cost effective? We begin by comparing the upgrade costs of the two approaches. The upgrade cost per ESLOC of the OMS demonstration platform for mission processor software and application interface/platform integration software is compared against lower and upper bounds from the upgrade costs of the historical non-OMS aircraft to determine if the OMS costs are indeed lower as theory suggests.[12] If it is found that the OMS costs are *outside* the bounds (more specifically, below the lowest value), there is reason for optimism that the OMS approach will result in lower costs. Each software category (Category 1 and 2) is analyzed independently, and then analyzed together (both categories combined). This is done deliberately as only six of the historical programs contain data for both categories of software. Table 2 summarizes the results.

Table 2. Cost per ESLOC comparison

Category	*OMS Cost*	*Historical (non-OMS)* *Lowest Value*	*Historical (non-OMS)* *Mean*	*Historical (non-OMS)* *Highest Value*
Cat 1: Mission Processor	**$30.78**	$51.36	$278.52	$667.40
Cat 2: Application Interface/Platform Integration	**$88.54**	$152.61	$949.45	$2,874.20
Cat 1–2 Combined	**$62.65**	$106.74	$421.36	$1,017.92

Table 2 shows that the OMS demonstration platform is less expensive than the analogous non-OMS historical programs. For Category 1 (mission processor), the non-OMS historical programs have a mean cost of $278.52 per ESLOC with lower and upper bounds of $51.36–$667.40. The associated cost per ESLOC for the OMS program, however, falls far outside the lowest value at only $30.78. The same result (OMS falls well below the lower bound) is true for the Category 2 analysis and the combined Category 1 and 2 analysis. These findings are promising, but we again caution the reader that they should be considered preliminary due to the small data set analyzed.[13]

While the analysis from table 2 is favorable to OMS, it only captures the cost *per unit* of ESLOC. The cost per ESLOC is important, but the *total cost* to upgrade an asset depends not only on the cost per ESLOC, but on the *total quantity* of ESLOC required as well. For example, consider a program where the traditional approach costs $10 per line of code and requires 100 lines ($10 * 100 = $1,000) while the OMS approach costs $9 per line of code and requires 200 lines ($9 * 200 = $1,800). In this example, despite OMS being cheaper on a per unit basis ($9 vs. $10), the overall cost of the OMS approach is more expensive ($1,800 vs $1,000). The key component is the total ESLOC used by the two approaches. As shown in the example, if OMS requires more ESLOC than the historical proprietary systems then it may be more expensive from a total cost aspect. To investigate this, the required total quantity of ESLOC in the OMS demonstration platform is examined against lower and upper bounds[14] for the required ESLOC in the historical non-OMS programs. Table 3 summarizes the result.

Table 3. Total ESLOC Lines Comparison

Category	*OMS Count*	*Historical (non-OMS)*	*Historical (non-OMS)*	*Historical (non-OMS)*
		Lowest Value	*Mean*	*Highest Value*
ESLOC	37,657	77,762	544,189	1,687,489

Not only did the OMS approach reduce the cost per line of ESLOC as previously shown in table 2, but as shown in table 3 the amount of ESLOC required to upgrade the asset was also significantly lower than the historical proprietary programs. The OMS ESLOC count of 37,657 falls well below the lowest value (77,762) for the historical programs. Based on the results of this analysis, it appears that OMS is the better approach in cost per line and at the same time is likely to require *less* coding effort (as measured in lines of code counts). In other words, it is *unlikely* that any OMS cost per ESLOC savings would be offset by vast increases in the quantity of ESLOC required in the OMS approach. The bottom-line is that the evidence in this exploratory analysis suggests the OMS approach should result in reduced costs through *both* lower cost per ESLOC and lower lines of code counts.

But what about initial investment costs to make the asset "OMS ready" in the first place? The above discussion compared costs of the upgrades themselves. One could argue that there is a barrier to entry cost (an initial investment) that may make OMS too expensive. Take personal computers circa 1990s as an analogy. At that time, Apple computer owners were required to buy everything from Apple.

Any monitor, printer, or mouse had to be designed and built by Apple to work with Apple's interfaces and operating system. As competition entered the market and HP, Dell, Gateway, Compaq, and countless others started making computer printers and other peripherals, the Apple owners missed out on all the savings of market competition. But for a one-time cost, they could abandon Apple and switch to Windows. For the remainder of the life of their machine, they could reap the benefits of competition and save money on their upgrades. In the same manner, the OMS comparison above is like updating to a different computer and operating system. Before upgrades can happen, the asset needs to be an "open system" or OMS ready. If the asset was designed from conception as an open system utilizing the OMS standards, this is a non-issue. But if it was not, then there is a cost to make it an open system. This is a one-time nonrecurring cost covering the useful life of the asset.

Thus, to complete the cost analysis, we consider the initialization cost in addition to the previously discussed upgrade costs. The initial investment cost to make the asset OMS ready was $5.4M. We compare this to the savings projected from the most optimistic scenario from the historical proprietary system upgrades. If the proprietary upgrade is assumed to require only 77,762 ESLOC (the lower bound table 3) at a cost per ESLOC of $106.74 (the lower bound from table 2) the proprietary upgrade would still exceed the cost of an OMS upgrade by over $5.9M. This means the OMS approach remains $0.5M cheaper ($5.9M- $5.4M = $0.5) when including both the one-time nonrecurring investment cost and the recurring cost of the upgrade itself. It is important to note that this scenario only assumes one upgrade in the life of the asset. If more than one upgrade occurred, the savings from the OMS approach would be even larger. Under the same conservative assumptions, a second upgrade would realize the full $5.9M in savings for the OMS approach plus the previous $0.5M for a total of $6.4M. All subsequent upgrade instances would accrue an additional $5.9M savings in perpetuity until the end of the useful life of the asset under these assumptions. Again, these are tentative findings due to the limitation of the data, but they utilize the most conservative assumptions to give the historical proprietary approach as much credit as possible.

While the previous analyses focused on cost-efficiencies, schedule is also an important consideration. The length of time to complete an upgrade must be considered as it directly affects combat capability. Therefore, the Research and Development (R&D) integration time of the OMS demonstration platform is compared against lower and upper bounds of the R&D integration times of the historical programs. Table 4 summarizes the schedule results.

Table 4. Schedule comparison

Category	*OMS Months*	*Historical (non-OMS)* *Lowest Value*	*Historical (non-OMS)* *Mean*	*Historical (non-OMS)* *Highest Value*
Upgrade R&D Months	3	14	33.36	60
Initialization Months	9	N/A	N/A	N/A

The first row of table 4 is a direct comparison of the upgrade times. The OMS R&D integration time of three months was far below the historical proprietary programs lowest value of 14 months. While that is a direct comparison of upgrade times from the two approaches, what are the schedule impacts of first making an asset "OMS-ready" to conduct these upgrades? The second row of table 4 provides this information. The R&D time to make the OMS demonstration platform an open system was nine months. Combining the initialization time of nine months in conjunction with a three-month upgrade results in a total schedule of 12 months. This result is still below the historical proprietary programs' lower bound of 14 months. Similar to the initialization cost discussion, it is important to note that the nine months is a one-time event. The upgrades, however, may be numerous (i.e., recurring) throughout the life of the asset. The projected schedule savings for subsequent upgrade instances would be calculated with the data from row 1. These preliminary results, therefore, are favorable to the OMS approach even when the initialization schedule is included.

Limitations

Despite the promising findings of this study, important limitations do exist. While this OMS aircraft platform provided insight into the OMS possibilities, the conclusions drawn in this article are from a single platform and should be considered exploratory. One can be optimistic about the prospects of OMS, but definitive conclusions cannot be drawn. More data must be made available as more platforms are OMS-enabled to conclude that OMS provides significant cost and schedule savings over proprietary platforms.

The dearth of OMS data at this time prohibited statistical testing. Sufficient data was available to develop confidence intervals for the historical proprietary programs. However, confidence intervals could not be developed for the OMS approach because it only contained one data point.

Additionally, an underlying premise of the analysis was that as software coding efforts increase, costs increase. The regression analysis found positive correlation. However, the relationship is only considered moderately strong (see note 11).

Furthermore, the historical programs deemed analogous to the OMS aircraft platform in this article might not be all encompassing. The data set only included USAF and US Navy Acquisition Category I programs. Different systems across the DOD and systems under a different acquisition category could be deemed analogous to the OMS aircraft platform upgrades and provide a broader data set from which to test the OMS aircraft platform data.

Despite these limitations, the OMS demonstration platform represents a real-world asset that has undergone the OMS transformation and experienced several component upgrades. While the conclusions cannot be deemed definitive, they provide a glimpse into the potential future of avionics acquisition.

Conclusions

Although the data for OMS-enabled platforms are limited, early indications show that there are potential cost savings with OMS upgrades over the historical proprietary approach. The lower R&D times for OMS upgrades is also promising. Based on the findings in this exploratory analysis, OMS provide an avenue for rapid acquisition while they also lower the integration costs of upgrades on DOD platforms.

The implications for practitioners in the field are clear. With top-level support, practitioners can strive to develop more avionics upgrades that embrace the OMS approach. Gathering additional data through these new OMS efforts is necessary to validate the preliminary results shown here. Finding consistently reduced cycle times and lower costs should translate to more capability in the field. The future of OMS in the DOD shows great promise. More needs to be done to validate this promise. Now is the time to employ OMS and through subsequent data analysis determine whether they should be the preferred approach to avionics upgrades. ✪

Capt Michael J. Brown, USAF

Captain Brown (BS, USAFA; MS, Air Force Institute of Technology [AFIT]) is the deputy cost chief for the E-3 Sentry Airborne Warning and Control Systems Division at Hanscom AFB, Massachusetts.

Robert D. Fass, PhD

Dr. Fass (BA, MBA, University of New Mexico; PhD, New Mexico State University) is an assistant professor in the Air Force Institute of Technology [AFIT]) Department of Systems Engineering and Management.

Jonathan D. Ritschel, PhD

Dr. Ritschel (BBA, University of Notre Dame; PhD, George Mason University) is an assistant professor of cost analysis in the Air Force Institute of Technology [AFIT]) Department of Systems Engineering and Management.

Notes

[1] Don C. Winter, "Open Systems Architecture—A Boeing Enterprise Perspective," Report 0704-0188 (Arlington, VA: Boeing Open Systems Architecture Phantom Works, 2002), 1–14, https://apps.dtic.mil/.

[2] Office of the Deputy Assistant Secretary of Defense, Systems Engineering, "Modular Open Systems Approach," DOD.

[3] Richard Whittle, "Lt. Gen Otto Commits Air Force To More Open Mission Systems," *Breaking Defense*, 26 October 2015, https://breakingdefense.com/.

[4] Bill Filyner, "Open Systems Avionics Architectures Considerations," *IEEE AES Systems Magazine* 18, no. 9 (2003): 3–10.

[5] Filyner, "Open Systems Avionics Architectures Considerations," 3–10.

[6] Whittle, "Lt. Gen Otto Commits Air Force."

[7] Lockheed Martin, "Lockheed Martin Completes Sixth Open Mission Systems Flight," *PR Newswire*, 22 July 2015, https://news.lockheedmartin.com/.

[8] Whittle, "Lt. Gen Otto Commits Air Force."

[9] All data were collected in raw form in a standard Work Breakdown Structure format (compliant with Appendix B of Mil-Std-881D), and all costs were normalized to Base Year 2014.

[10] The equation is $\text{ESLOC} = \text{New}_{\text{SLOC}} + (.85 \times \text{Modified}_{\text{SLOC}}) + (.05 \times \text{Reused}_{\text{SLOC}})$.

[11] Regression result found a statistically significant positive correlation of 0.665 at the 95-percent confidence level.

[12] Table 2 shows the range and mean of the non-OMS platform data, along with the single OMS platform data point. Three additional analyses were conducted on this data. First, a 90-percent normally distributed confidence interval around the mean of the historical non-OMS upgrade costs was developed. The underlying assumption of normality in the historical data was tested and verified through the Shapiro-Wilk test and W/S test. The confidence interval bounds were $138.14–$418.89 for Category 1, $452.19–$1,446.17 for Category 2, and $146.77–$695.95 for both categories combined. The OMS cost was below the lower bound in all three of these instances. The second analysis constructed 95 percent prediction intervals on the non-OMS upgrades, utilizing each non-OMS upgrade as the X0. These prediction intervals were then compared to the OMS data point. Due to the small data set size and associated variability, the prediction intervals were very wide. The lower bound of the prediction interval crossed zero in all but one instance for Category 1, all but one instance for Category 2, and in all instances for both categories combined. The third analysis applied Chebyshev's Theorem, specifically examining the interval from -2 to 2 standard deviations from the mean. In Chebyshev's interval, this calculation is interpreted as at least 75 percent of the observations fall within the range. The interval for Category 1 was -$140.61–$697.64, -$768.19 –$2,665.09 for Category 2, and both categories combined were -$246.23–$1,088.94. The OMS cost was within the bounds in all three instances. These three analyses demonstrate the limits of statistical analysis with this very small data set.

[13] We ran an additional analysis here, making use of recommendations from the *Joint Agency Cost Risk and Uncertainty Handbook*, 16 September 2014, 31–33, https://www.ncca.navy.mil/. Essentially, we ran Monte Carlo simulations (10,000 iterations), using triangular distributions on the historical data. The historic low values were set to the 0.08 fractile, the mean was set to 0.50, and the historic high values were set to the 0.77 fractile. The resulting cumulative distribution functions for the mission processor, interface/integration, and combination categories (OMS system actuals) were 1.36, 1.29, and 2.13 percent respectively. Again, with caveats related to our small

sample size in mind, we can interpret these percentages as very low probabilities that we would expect the cost of a system to be as low as the OMS actuals were. Details on this technique are available from the authors upon request.

14. Table 3 shows the range and mean of the non-OMS platform data, along with the single OMS platform data point. The same three analyses discussed in note 12 are conducted on the ESLOC data. Results mirror the findings of note 12, where the OMS ESLOC count falls outside the lower bound of the confidence interval but is contained within the prediction interval and Chebyshev's interval.